PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04N

A2

(11) International Publication Number: WO 98/57489

(43) International Publication Date: 17 December 1998 (17.12.98)

(21) International Application Number: PCT/US98/11485

(22) International Filing Date: 8 June 1998 (08.06.98)

(30) Priority Data:
08/872,480 9 June 1997 (09.06.97) US

(71) Applicant: METALITHIC SYSTEMS, INC. [US/US]; Suite 206, 3 Harbor Drive, Sausalito, CA 94965 (US).

(72) Inventors: O'REILLY, James, M.; Suite 206, 3 Harbor Drive, Sausalito, CA 94965 (US). HIGEN, Daryl; 2280 Paradise Drive, Tiburon, CA 94920 (US).

(74) Agents: KORN, Martin et al.; Locke Purnell Rain Harrell, Suite 2200, 2200 Ross Avenue, Dallas, TX 75201 (US).

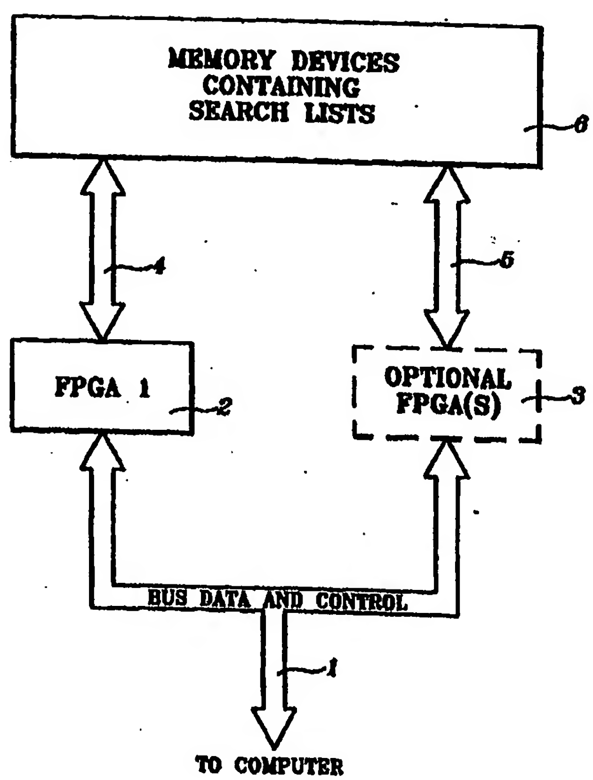
(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: MODULAR SYSTEM FOR ACCELERATING DATA SEARCHES AND DATA STREAM OPERATIONS

(57) Abstract

Using a modular reconfigurable Logic architecture coupled with a dense and flexible packaging scheme, it is possible to develop an engine with very high search speed and capable of complex search operations or data stream operations. This technology has great applicability in the areas of data mining, recognition of continuous speech, automated translation and image analysis/processing.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

MODULAR SYSTEM FOR ACCELERATING DATA SEARCHES AND DATA STREAM OPERATIONS

TECHNICAL FIELD OF THE INVENTION

This invention generally relates to integrated circuit computing devices and to computer system designs. More specifically, it relates to a combination of memory devices and Field-Programmable gate Arrays, together forming a Module which can be used to accelerate list processing functions such as database searches, speech recognition, speech or text translation, data stream transformation as in video or image editing, or routing of communications messages.

BACKGROUND OF THE INVENTION

Most computers use a simple architecture of a single memory sub-system and a single processor or set of processors accessing that memory. As a result, many systems are unable to perform so-called data-stream operations efficiently, and, are limited by the memory bandpass of the system in achieving total performance.

This limits the ability of the conventional computer to handle large data sets (data-streams) at an adequate level of performance. Such performance limitation prevents deployment of, for example, speech to text and automated translation systems.

To overcome this issue, and to provide the capability that demanding data-stream operations place on a system, it is necessary to: (a) increase memory bandwidth, (b) increase compute power by parallel processing, and (c) define a compute/comparison engine running at very high speed.

The invention described herein addresses each of these issues and achieves a dramatic performance boost for these type of operations. It provides a means to increase memory bandwidth by adding semi-autonomous Modules, it adds several layers of parallelism in computing, data transforming or comparison. The architecture is designed for the specific set of tasks required, but, since it is based on Reconfigurable Logic, the electronic circuits on which it is based can be rapidly modified at any given point in time to be optimal in configuration for the task at hand.

SUMMARY OF THE INVENTION

Using a combination of memory devices and field Programmable gate Arrays (FPGA), it is possible to build a modular system for accelerating data searches to much higher levels of performance than can bed realized with a simple computer system. This system achieves performance improvement in several ways.

First, the use of a combination of memory devices and FPGA's allows a much higher effective memory access rate than conventional computer architectures, with total memory bandpass increasing as each new module is added.

Second, because the architecture is independent of any computer structure the speed of access of each module to its memory component can be optimized to take advantage of special high-speed memory access modes such as fast page mode.

Third, the comparisons and other functions take place at hardware speeds, since the modular architect described herein does not require the structure of program steps typically seen in a conventional computer system.

Fourth, complex comparisons that involve logical or mathematical transforms of either the Search List data or the Search Target data can occur in a pipelined stream of hardware operations, permitting very sophisticated and complex operations, which, again, occur at hardware speeds.

The memory devices and FPGA's that make up a module can be packaged together in a variety of ways. Packaging choices include placing the elements on an adapter card that plugs into the computer bus, or into a special bus dedicated to the search functions. To achieve a dense and

flexible packaging means, the combination of devices that makes up a module can be packaged onto a SIMM, DIMM or similar plug-in module. This permits the modules to be packed closely together, and allows the system designer choices as to whether the module is inserted into the sockets on the main processor board, or into sockets on a separate adapter card, where the constraints of the computer memory system can be ignored.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and for further advantages thereof, reference is now made to the following Description of the Preferred Embodiments taken in conjunction with the accompanying Drawings in which:

Figure 1 identified the basic structure of this invention, showing the connection of the various elements and optional elements and the function of the interconnections.

Figure 2 is an alternative method of connecting the elements together.

Figure 3 shows the functional content of the FPGA(s).

Figure 4 identified the incorporation of a processor or programmable controller element into the FPGA(s).

Figure 5 demonstrates how parallel function is achieved within a FPGA(s).

Figure 6 shows the preferred packaging scheme.

Figure 7 shows a scheme for connection of multiple Modules.

Figure 8 shows the connection of multiple Modules to operate on a large number of characters in parallel.

Figure 9 shows how multiple parallel comparisons are made using the same data lists.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The most basic embodiment of this invention is shown in Figure 1. Data and control (such as timing signals and address signals) are transferred from the computer on the Bus Data and Control lines 1 and into the FPGA 2 and/or additional optional FPGA's 3.

In the FPGA's 2,3, the data and control signals are modified to generate the Modified Data and Control signals 4 which are used to control the actions and contents of the memory Devices 6. Such modifications may include: 1) generating different address values than the one sent by the computer, 2) generating the required control and address values to permit reading data from the Memory Devices 5 to compare with values loaded into the FPGA(s) 2,3.

There are alternative methods of connecting the Memory Devices 5 to the FPGA(s) 2,3. Figure 2 shows one such alternative method, where the same Modified Data and Control 4 are shared by all the FPGA's 2,3, as opposed to the method shown in Fig. 1 where different Modified Data and Control 4,5 go to each FPGA 2,3. Such alternative methods are reconfigurable by connection of different logic in the FPGA(s) 2,3. This allows different operations to be performed in the several FPGA(s) 2,3 in the case of Figure 1, while the method of Figure 2 permits operation on the same or related data.

In Figure 3, the elements within the FPGA(s) 2,3 are detailed. Here is shown how data from the Memory Devices containing Search Lists 6 are moved into and from the FPGA(s) 2,3 with some combination of Transforms 7, Math Functions 8 and Comparators 9 being used to modify and/or examine the data. For clarity, only one of each such Transform 7, Math Function 8 or Comparators 9 is shown. A typical embodiment

might have several of each, in any order, connected to operate consecutively on data. The control logic 10 manages the sequence of events inside FPGA(s) 2,3.

To effect a typical search, data constituting Search Lists are placed into the Memory Devices 6. Depending on application of the embodiment, this might be done by using rapidly reprogrammable Memory Devices, such as dynamic Random Access Memory (DRAM) or Static Random Access Memory (SRAM), semi-static memory devices that are typically programmed infrequently or only at the time of initial assembly of the embodiment, such as FLASH memory or electrically Erasable Programmable read-Only Memory (EEPROM) or one-time programmable Memory Devices such as Mask-Programmable Read-Only Memory (ROM).

Following the placement of the Search List data, the FPGA(s) 2,3 are re-programmed from an initial start-up state to be able to manipulate the Search List data now stored in the Memory Devices 6. Such manipulations are effected by placing the functional elements, Transforms 7, Math/Logic Functions 8 and Comparators 9 in any sequence or quantity to act upon selected data elements of the Search List data.

A data item (Search Target) to be compared against the Search List is placed into FPGA's 2,3. Data from the Search List are then moved, data item by data item, into the FPGA(s) 2,3, where the instantiated Transforms 7 and Math/Logic Functions 8 operate on said Search List data item, following which said modified data item is compared with the Search Target inside Comparator 9. If a match is found between the Search Target and the Search List data item, the Control Logic 10 then informs the computer that a match has been found. Said Control Logic may be programmed to continue on for additional matches to the same Search

target data, or re-loaded with new Search Target data, and the Search List and FPGA contents may be changed at any time as required to optimize performance.

Figure 4 extends the concept described above to allow a programmable controller or processor 11 to be instantiated into the FPGA. This permits much greater flexibility in operation, since the sequence of hardware events, and the interaction of the module(s) with a host computer are capable of being modified.

Figure 5 shows an extension of the embodiment where multiple search operations occur in parallel. This is realized by instantiating sets of the various Transforms 7, Math/Logic Functions 8 and Comparators 9 into FPGA(s) 1 (etc.) and loading either the same or different Search Target data elements to correspond with each such set, which may contain different sizes and types of transforms 7, functions 8 and comparators 9. The operation in such multiple search mode follows the sequence above for a single search path, with the set of Search Target data items being compared with either the same Search List data items, as (optionally) modified by the (possibly different) set of Transforms 7 that are applied in each search path, or with different Search List data items, similarly modified.

The preferred packaging scheme (Figure 6) for the Modules is the SIMM. In this means, Memory Devices 6 and FPGA's 1,2 are mounted on one of several industry-standard form-factor boards to make a Module. This permits a very dense package, taking up a small physical space, and advantageously is supported by many computer systems. Alternative packaging schemes include the industry standard PCM-CIA bus card, the DIMM card, the small footprint PCI card and many other standard form factors.

In a typical application, several Modules 16 will be mounted together to achieve modular increments of power. Figure 7 shows such a configuration. Note that each Module shares the Data and Control signals to the computer. This permits each Module 16 to be loaded with Search Data, Search Target and control information, and to communicate with the computer, while allowing the autonomous parallel operation of the Modules 16 during the searching or modifying of data.

The Modules 16 can also be connected in such a way as to communicate with each other. This permits comparison of very wide data elements, which might be useful in image or speech processing, for example. Figure 8 shows a means where this might be achieved by sharing the computer Data and Control Box 1, which is connected to all of the Modules, as an intercommunication path 12 between each Module. Determination of the success or otherwise of the search or data modification operations can be realized by either the computer system or a specially programmed Module 13.

Another method of using the Module architecture, shown in Figure 9, is to build several parallel search or transform paths in each Module. This can be done within a single FPGA, as shown, or within multiple FPGA's mounted on the same module and sharing the same data. This method has the benefit that different transforms, mathematical operations or comparison methods can be deployed in parallel, to act on the same data, or, if appropriate, different data, as required. This allows, in some circumstances, for a large multiplication of performance of the modular system.

CLAIMS:

1. A data processing module adapted to be connected to a computer for use with a computer, the computer including a memory for storing data, the module comprising:

a module memory for storing data; and

a programmable logic device connected to said module memory and adapted to be connected to the computer for receiving data stored in said module memory and the computer memory for processing data.

2. The module of Claim 1 wherein said programmable logic device includes a comparator for determining whether data stored in the computer memory is stored in said module memory.

3. The module of Claim 1 wherein said programmable logic device is programmable by data stored in said module memory for processing data stored in said module memory.

4. The module of Claim 1 wherein said module memory includes a random access memory device.

5. The module of Claim 1 wherein said module memory and said programmable logic device are mounted on a single in-line memory module having terminals for connection to the computer.

6, A data processing system for use with a computer, the computer including a memory for storing data, the system comprising:

a plurality of data processing modules, adapted to be connected to the computer, each of said modules including:

a module memory for storing data; and

a programmable logic device connected to said module memory and adapted to be connected to the computer for receiving data stored in said module memory and the computer memory; and

such that said plurality of data processing modules simultaneously process data stored in each of said module memories and the computer memory.

7. The system of Claim 6 wherein said programmable logic devices include a comparator for determining whether data stored in the computer memory is stored in said module memories.

8. The system of Claim 6 and further including:

means for transferring data between said plurality of data processing modules.

9. The system of Claim 6 wherein ones of said programmable logic devices perform comparisons on said data stored in said module memories.

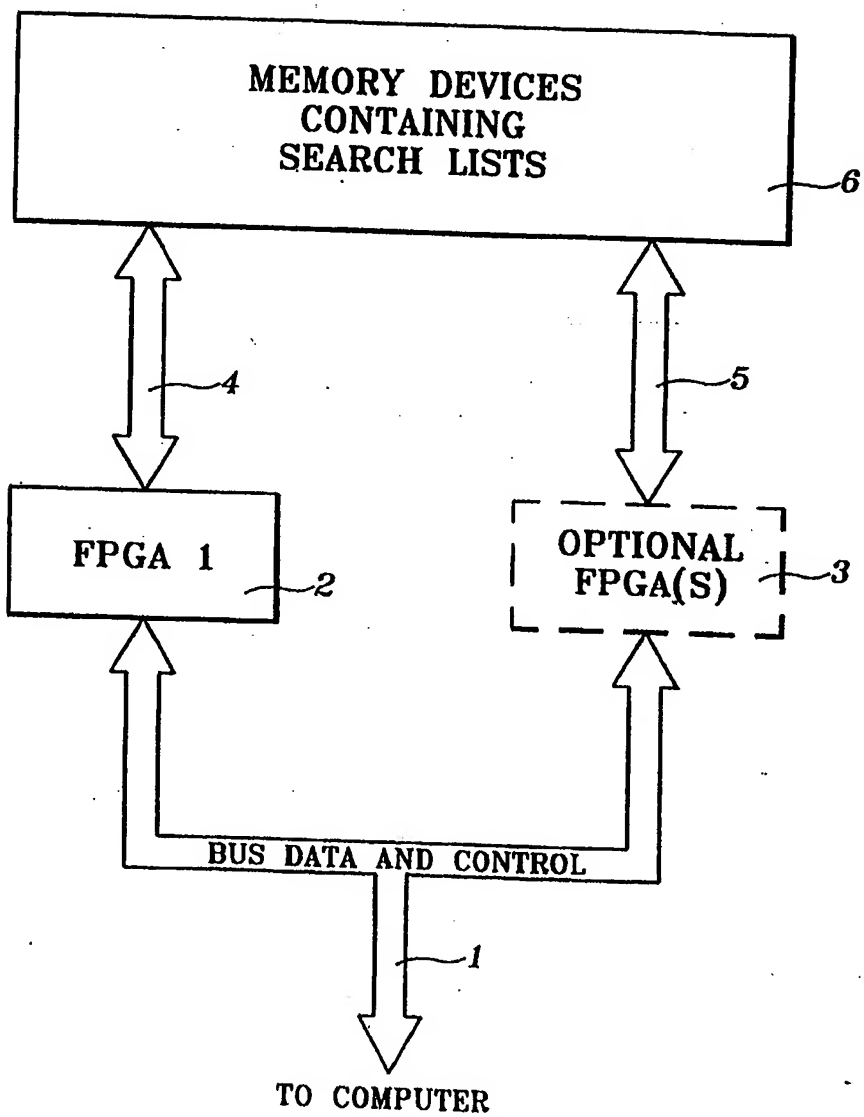
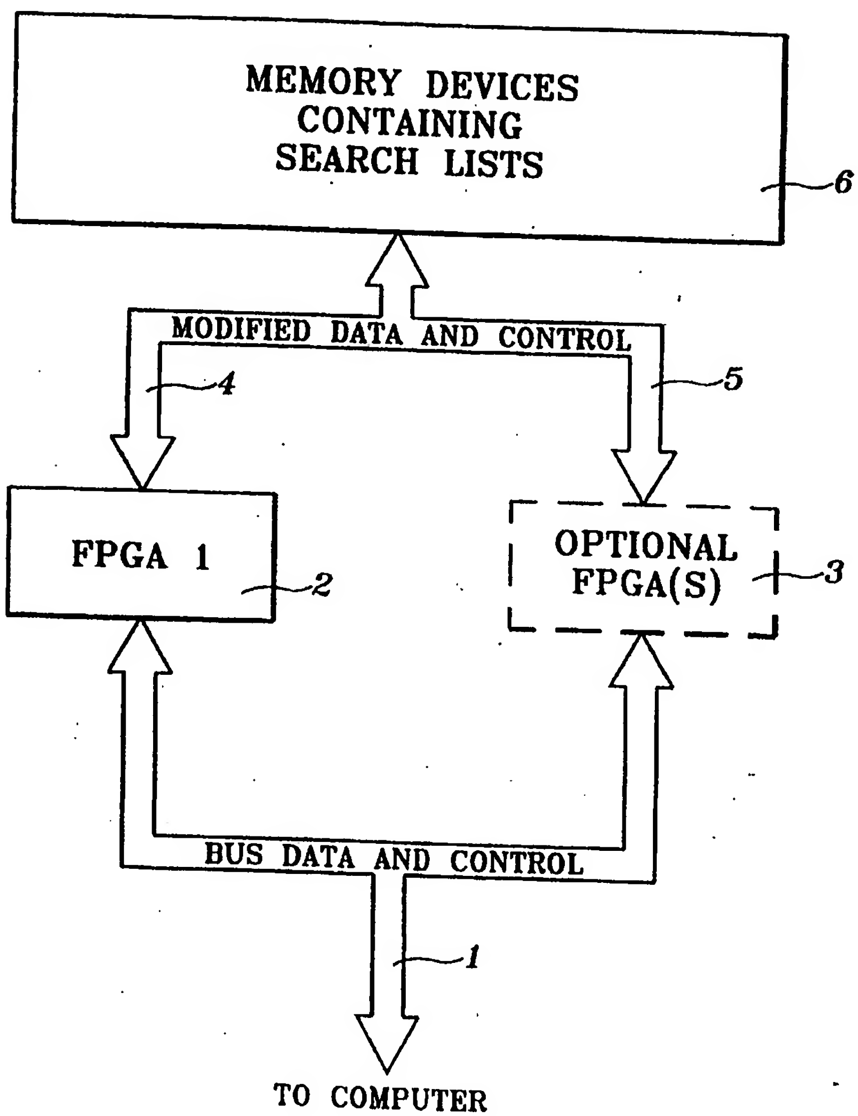

MEMORY DEVICES
CONTAINING
SEARCH LISTS
6
4
5
FPGA 1
2
OPTIONAL
FPGA(S)
3
BUS DATA AND CONTROL
1
TO COMPUTER

FIG. 1



MEMORY DEVICES CONTAINING SEARCH LISTS
6
MODIFIED DATA AND CONTROL
4
5
FPGA 1
2
OPTIONAL FPGA(S)
3
BUS DATA AND CONTROL
1
TO COMPUTER

FIG. 2

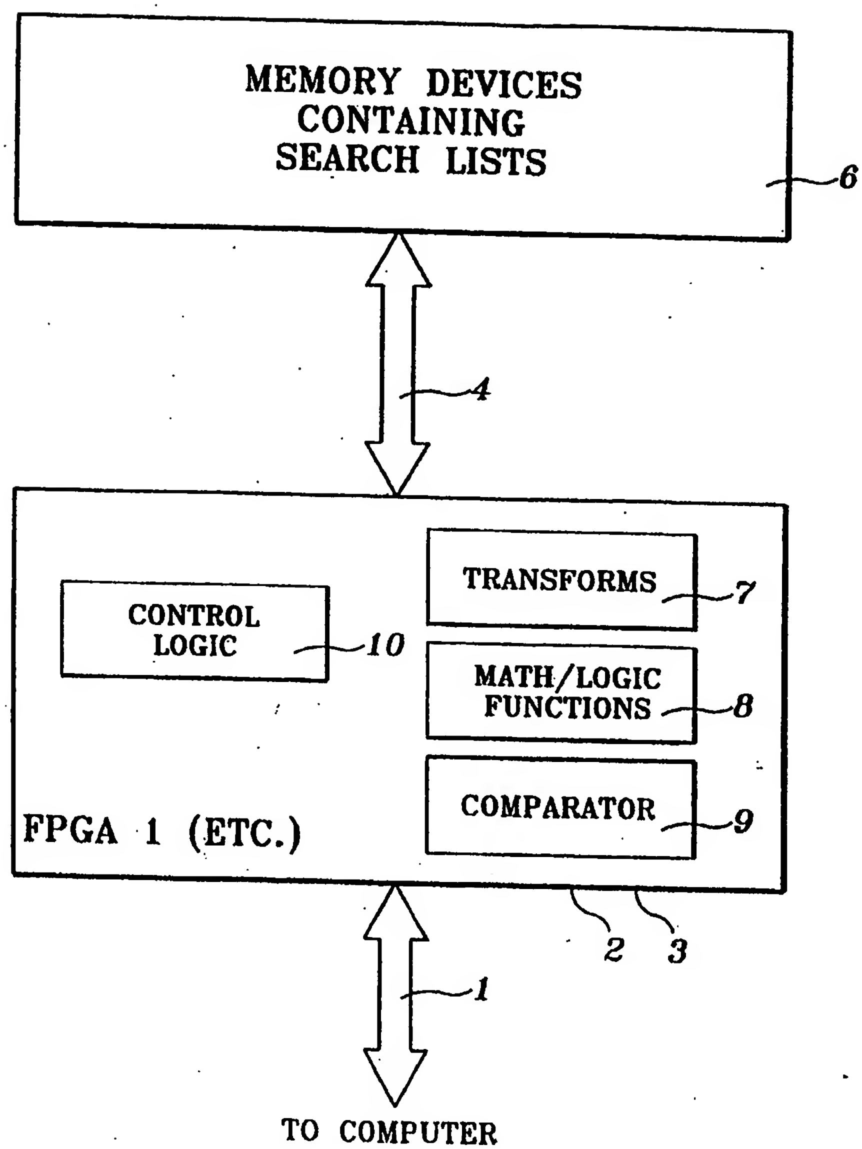

MEMORY DEVICES
CONTAINING
SEARCH LISTS
6
4
TRANSFORMS
7
CONTROL
LOGIC
10
MATH/LOGIC
FUNCTIONS
8
COMPARATOR
9
FPGA 1 (ETC.)
2
3
1
TO COMPUTER

FIG. 3

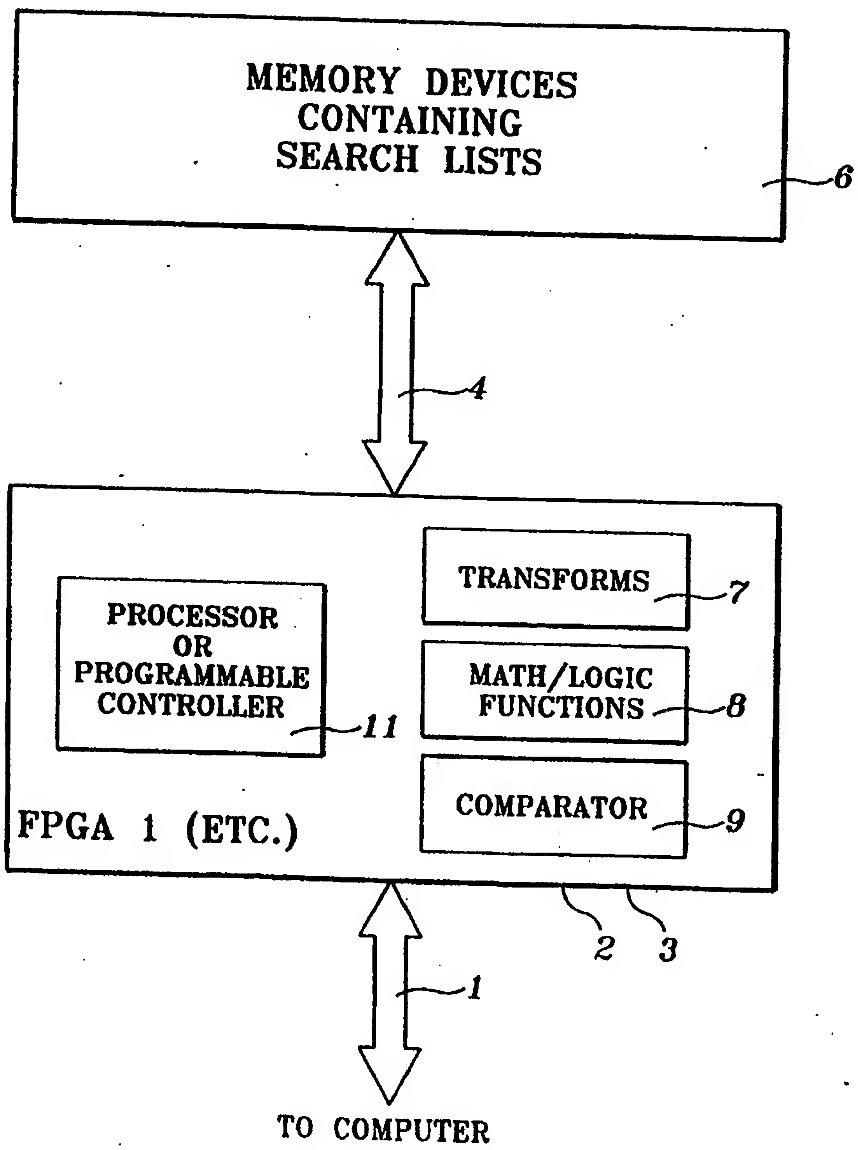

MEMORY DEVICES CONTAINING SEARCH LISTS
6
4
PROCESSOR OR PROGRAMMABLE CONTROLLER
11
TRANSFORMS
7
MATH/LOGIC FUNCTIONS
8
COMPARATOR
9
FPGA 1 (ETC.)
2
3
1
TO COMPUTER

FIG. 4

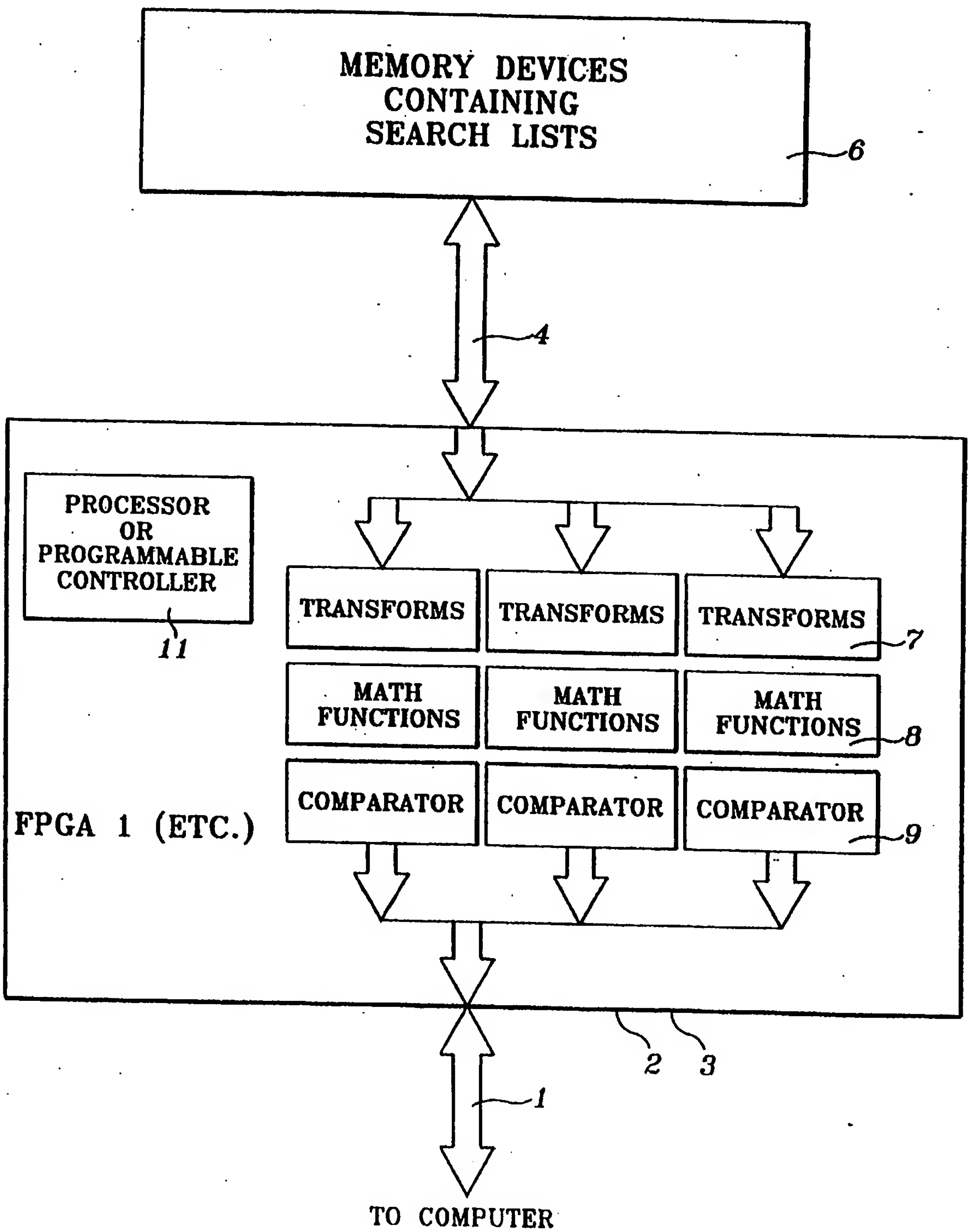

MEMORY DEVICES
CONTAINING
SEARCH LISTS
6
4
PROCESSOR
OR
PROGRAMMABLE
CONTROLLER
11
TRANSFORMS
TRANSFORMS
TRANSFORMS
7
MATH FUNCTIONS
MATH FUNCTIONS
MATH FUNCTIONS
8
COMPARATOR
COMPARATOR
COMPARATOR
9
FPGA 1 (ETC.)
2
3
1
TO COMPUTER

FIG. 5

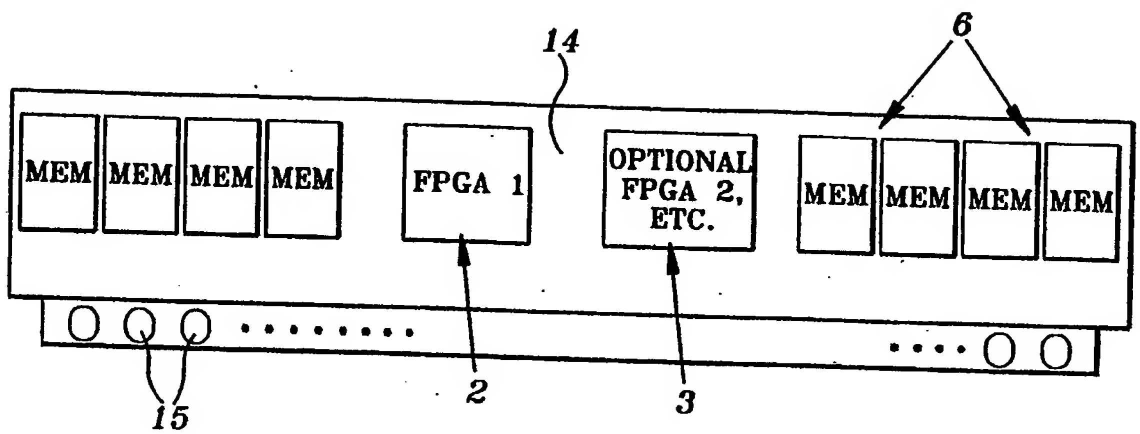

14
6
MEM
MEM
MEM
MEM
FPGA 1
OPTIONAL FPGA 2, ETC.
MEM
MEM
MEM
MEM
15
2
3

FIG. 6

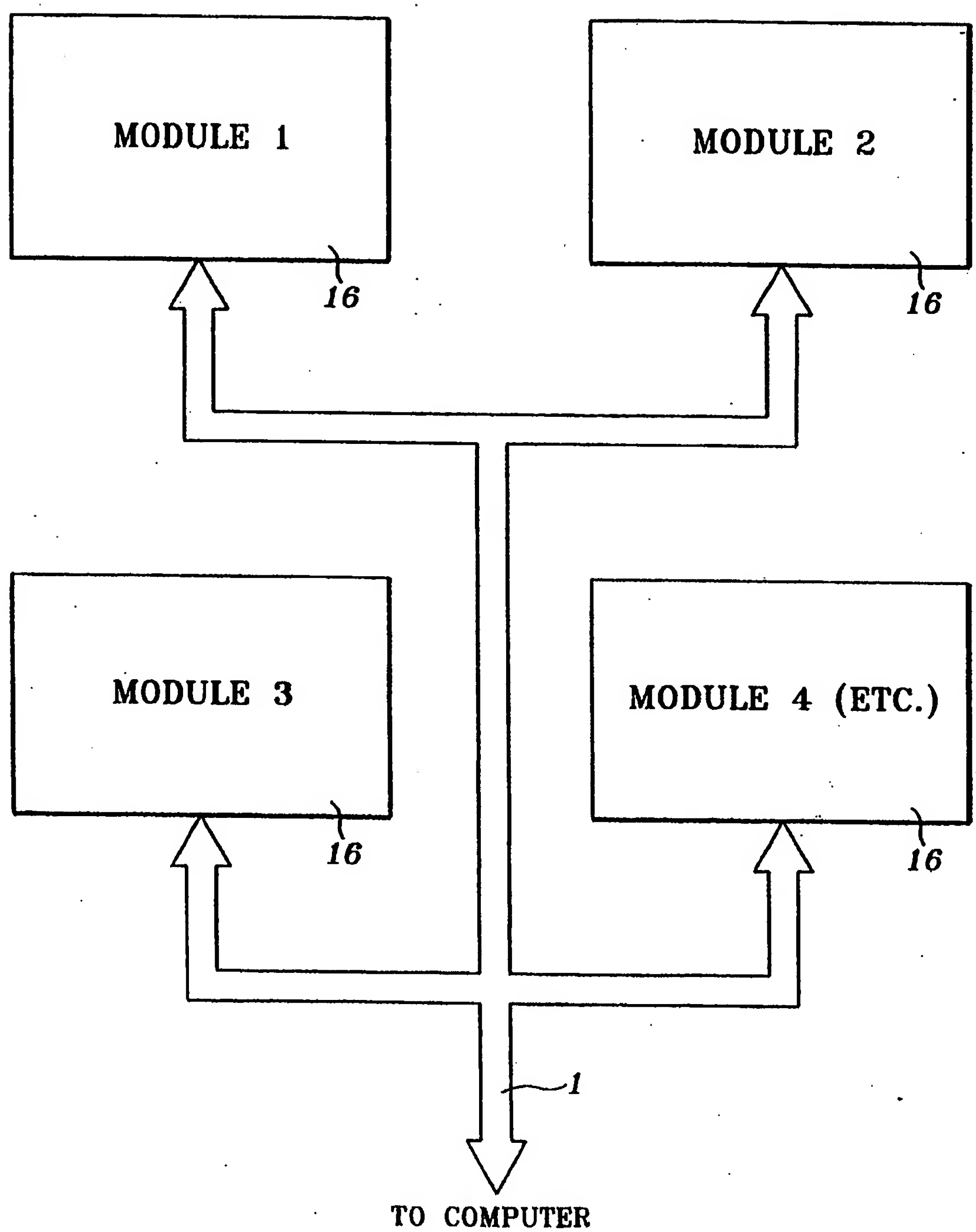

MODULE 1
MODULE 2
16
16
MODULE 3
MODULE 4 (ETC.)
16
16
1
TO COMPUTER

FIG. 7

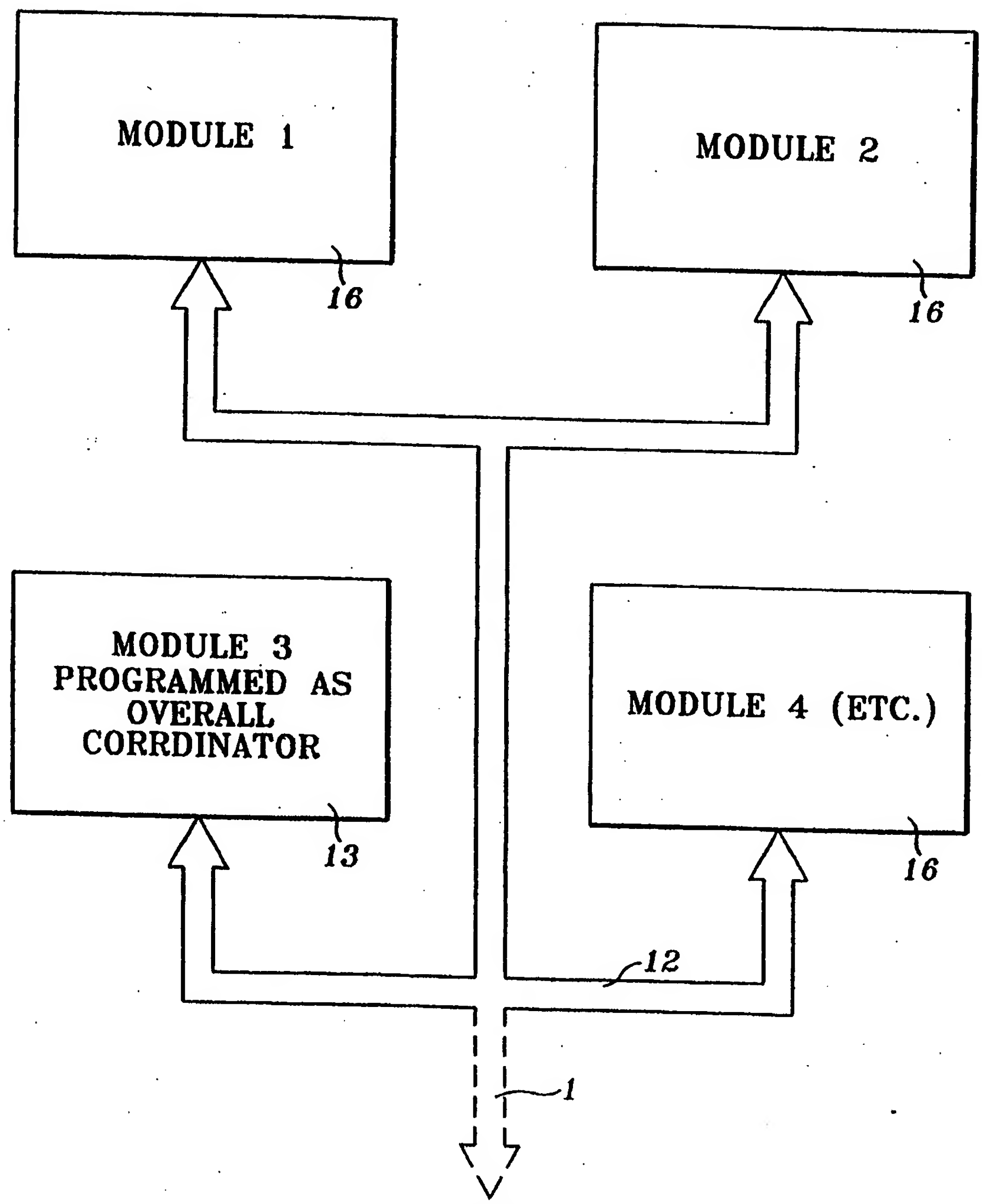

MODULE 1
16
MODULE 2
16
MODULE 3
PROGRAMMED AS
OVERALL
CORRDINATOR
13
MODULE 4 (ETC.)
16
12
1
TO COMPUTER

FIG. 8

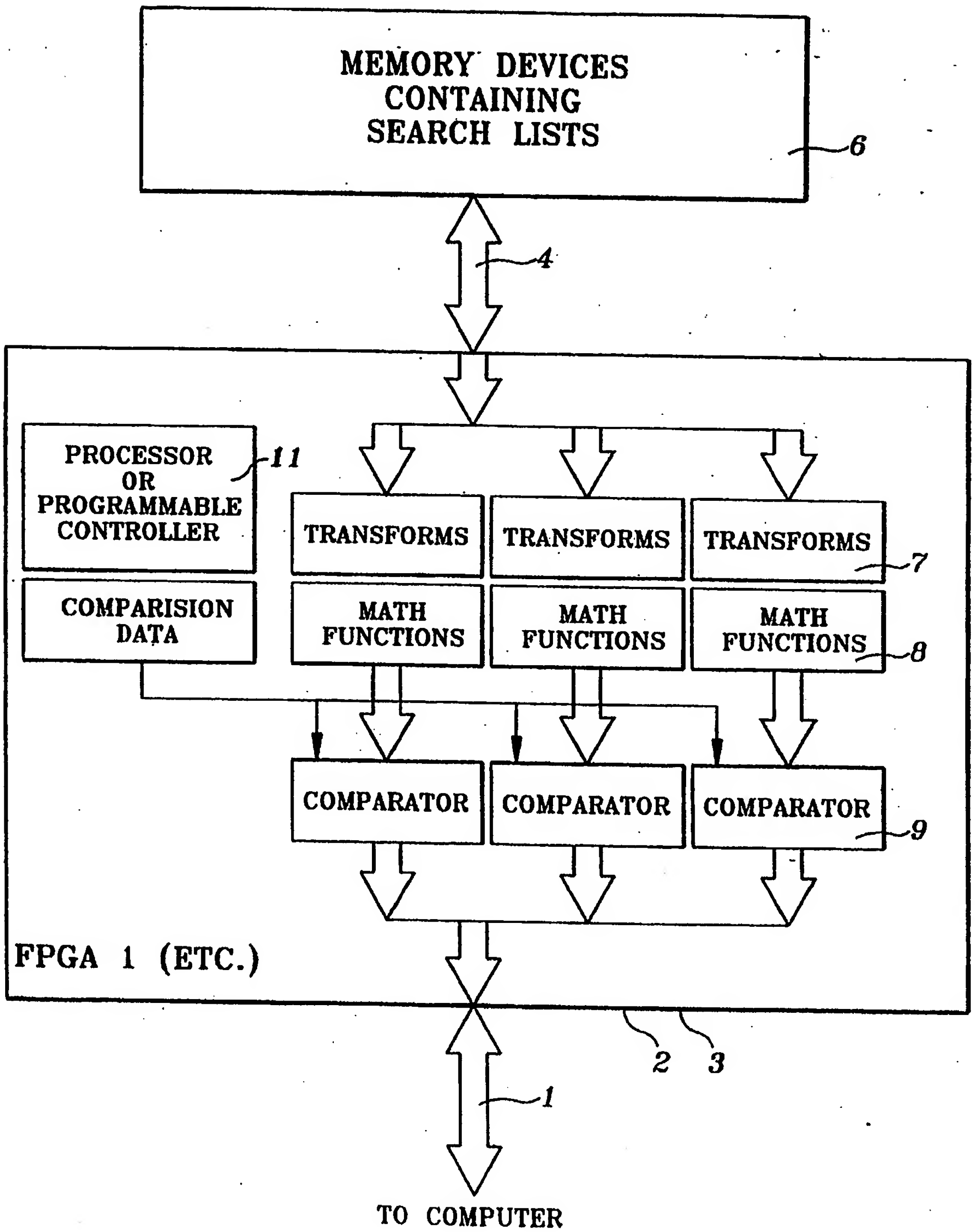

MEMORY DEVICES CONTAINING SEARCH LISTS
6
4
PROCESSOR OR PROGRAMMABLE CONTROLLER
11
TRANSFORMS
TRANSFORMS
TRANSFORMS
7
COMPARISION DATA
MATH FUNCTIONS
MATH FUNCTIONS
MATH FUNCTIONS
8
COMPARATOR
COMPARATOR
COMPARATOR
9
FPGA 1 (ETC.)
2
3
1
TO COMPUTER

FIG. 9